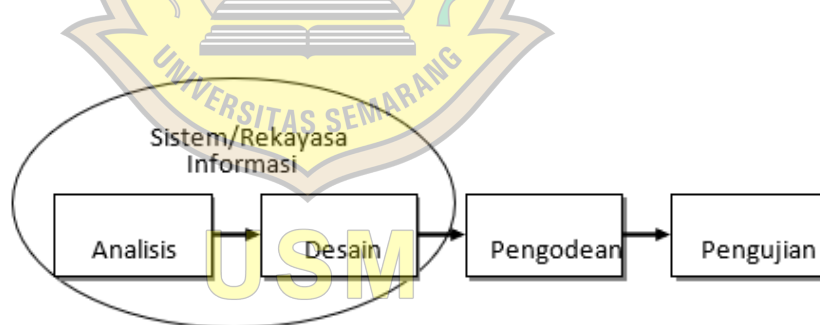


BAB III

LANDASAN TEORI

3.1 Konsep Dasar Metode *Waterfall*

Metode pengembangan perangkat lunak diperlukan dalam setiap perancangan sebuah sistem. Dengan metode yang tepat, pembuatan sistem atau aplikasi diharapkan berjalan efektif dan efisien serta memenuhi kualitas dan kebutuhan yang diinginkan. Menurut Sukanto dan (Shalahuddin, 2015), “Model air terjun(*waterfall*) menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*). Tahapan metode *waterfall* dapat dilihat pada gambar 3.1.



Gambar 3. 1 Tahapan metode waterfall

Adapun metode *waterfall* menurut (Agustini, 2017) yaitu:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara insentif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat-perangkat lunak seperti apa yang dibutuhkan oleh user. . Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional serta memastikan bahwa bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) dan pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

3.2 Konsep Dasar Sistem Informasi

3.2.1 Pengertian Sistem

Menurut (Mulyani, 2016) menyatakan bahwa sistem bisa diartikan sebagai sekumpulan subsistem, komponen ataupun elemen yang saling

bekerja sama dengan tujuan yang sama untuk menghasilkan output yang sudah ditentukan sebelumnya.

Berdasarkan pengertian para ahli di atas maka dapat disimpulkan bahwa sistem adalah serangkaian elemen yang berkaitan dengan jaringan kerja yang saling berinteraksi satu dengan yang lainnya dan bertujuan untuk menangani dan menyelesaikan serangkaian kegiatan yang terjadi secara rutin dan berulang-ulang. Serangkaian elemen tersebut mempunyai arti bukan hanya elemen yang tampak secara fisik tetapi juga hal-hal yang bersifat abstrak atau konseptual seperti misi, pekerjaan, dan sebagainya.

3.2.2 Karakteristik Sistem

Pembuatan sistem harus memahami ciri-ciri atau karakteristik yang terdapat pada sekumpulan elemen yang ada, sebagai dasar pertimbangan dalam pembuatan sistem. Adapun karakteristik dari sistem yang dimaksud (Hutahaean, 2015) yaitu:

- 1) **Komponen**

Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

- 2) **Batasan Sistem (*Boundary*)**

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

- 3) **Lingkungan Luar Sistem (*Environment*)**

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem mempengaruhi operasi sistem disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4) Penghubung Sistem (*Interface*)

Media menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi integrasi sistem yang membentuk satu kesatuan.

5) Masukan system (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6) Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem.

7) Pengolah Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8) Sasaran Sistem (*objective*)

Suatu sistem mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

3.2.3 Pengertian Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya (Hutahaean, 2015) Sedangkan

menurut (Mulyani, 2016) bahwa informasi merupakan data yang sudah diolah yang ditujukan untuk seseorang, organisasi ataupun siapa saja yang membutuhkan.

Berdasarkan dari pendapat di atas maka dapat diambil kesimpulan bahwa informasi merupakan sekumpulan data yang telah diolah menjadi bentuk yang lebih berguna, menggambarkan suatu kejadian yang nyata dan dijadikan sebagai alat bantu untuk pengambilan keputusan bagi penggunanya.

Informasi yang berkualitas memiliki kriteria-kriteria tertentu. Menurut Romney dan Steinbart dalam (Mulyani, 2016) mengemukakan bahwa kriteria informasi yang terdiri dari:

1. Relevan

Informasi harus bisa dikatakan relevan apabila informasi yang termuat di dalamnya dapat mempengaruhi keputusan pengguna dengan membantu mereka mengevaluasi peristiwa masa lalu atau masa kini, dan memprediksi masa depan, serta menegaskan atau mengoreksi hasil evaluasi mereka di masa lalu.

2. Andal

Informasi harus bebas dari pengertian yang meyesatkan dan kesalahan material, menyajikan setiap fakta secara jujur, serta dapat diverifikasi.

3. Lengkap

Informasi disajikan selengkap mungkin, yaitu mencakup semua informasi yang dapat mempengaruhi pengambilan keputusan.

4. Tepat waktu

Informasi disajikan tepat waktu sehingga dapat berpengaruh dan berguna dalam pengambilan keputusan.

5. Dapat dipahami

Informasi disajikan dalam bentuk serta istilah yang disesuaikan dengan batas pemahaman para pengguna.

6. Dapat diverifikasi

Informasi yang disajikan dapat diuji, dan apabila pengujian dilakukan lebih dari sekali oleh pihak yang berbeda, hasilnya tetap menunjukkan simpulan yang tidak berbeda jauh.

7. Dapat diakses

Informasi yang tersedia pada saat dibutuhkan dan dengan format yang dapat digunakan.

3.2.4 Pengertian Sistem Informasi

Menurut (Hutahaean, 2015) mengemukakan bahwa sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Menurut Lucas dalam (Djahir, 2015) mengemukakan bahwa sistem informasi adalah suatu kegiatan dari prosedur-prosedur yang diorganisasikan, bilamana dieksekusi akan menyediakan informasi untuk mendukung pengambilan keputusan dan pengendalian di dalam organisasi.

Berdasarkan dari pendapat di atas maka dapat diambil kesimpulan bahwa sistem informasi merupakan gabungan dari manusia, *hardware*, *software*, jaringan komunikasi dan data yang saling berinteraksi untuk menyimpan, mengumpulkan, memproses, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dalam suatu organisasi.

3.2.5 Pengertian Database

Menurut (Andaru, 2018), database adalah tempat media penyimpanan data kita dalam membuat sebuah program yang berisikan *table*, *field*, dan *record* yang diselimuti namanya DBMS (*Database Management System*).

Menurut (Indrajani, 2015), basis data adalah kumpulan data saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi.

Berdasarkan pendapat di atas dapat disimpulkan bahwa database adalah sekumpulan data yang terkomputerisasi yang berisikan *table*, *fields*, dan *record*.

3.3 Analisa Dan Perancangan Sistem

3.3.1 Analisa Sistem

Analisa sistem adalah penguraian suatu sistem informasi yang sudah utuh ke dalam bagian-bagian komponennya dengan tujuan dapat mengidentifikasi dan mengevaluasi berbagai macam permasalahan maupun hambatan yang terjadi pada sistem sehingga nantinya dapat dilakukan perbaikan atau pengembangan (Jermias, 2016).

3.3.2 Perancangan Sistem

Perancangan adalah langkah pertama dalam fase pengembangan rekayasa produk atau sistem. Perancangan itu adalah proses penerapan berbagai teknik dan prinsip yang bertujuan untuk mendefinisikan sebuah peralatan, satu proses atau satu sistem secara detail yang membolehkan dilakukan realisasi fisik (Nadeak, 2016).

3.4 Alat Bantu Perancangan Sistem

3.4.1 Unified Modeling Language (UML)

Menurut (Shalahuddin, 2015) “UML (*Unified Modelling Language*) adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasikan objek.

Sedangkan menurut (Oktafianto, 2016) “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung”.

Jadi UML (*Unified Modeling Language*) dapat diartikan sebagai Bahasa visual untuk menggambarkan definisi-definisi tentang *requirement*, membuat analisis dan desain serta menggambar arsitektur dalam pemrograman berorientasikan objek dengan menggunakan teks-teks pendukung.

3.4.2 Pemodelan Pemrograman Berorientasi Objek

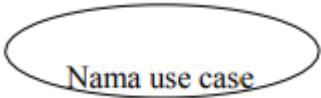

1. Use case Diagram

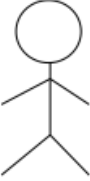
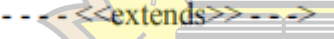
Menurut (Shalahuddin, 2015) *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal pada *use case* yaitu pemodelan apa yang disebut Aktor dan *use case*.


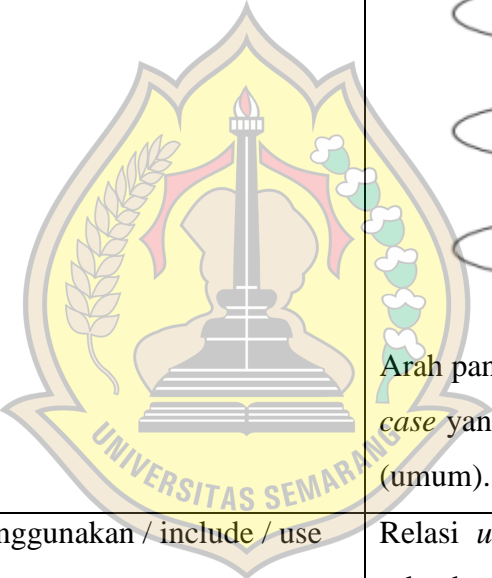
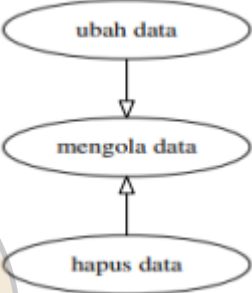
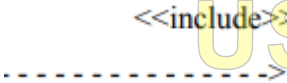
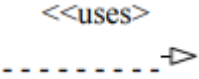
- 1) Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- 2) *Use case* merupakan fungsional yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit atau aktor.


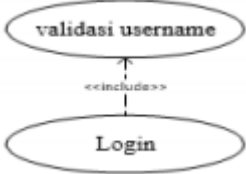
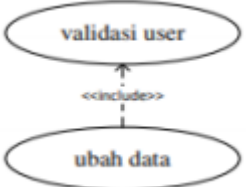
Tabel 3. 1 Simbol Use Case Diagram

Sumber : (Shalahuddin, 2015)

NO	SIMBOL	DESKRIPSI
1.	<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
2.	<p>Assosiasi / <i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

3.	<p>Aktor / <i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
4.	<p>Exstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>

5.	<p>Generalisasi / <i>generalization</i></p>  	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasi (umum).</p>
6.	<p>Menggunakan / <i>include</i> / <i>use</i></p>  	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Ada dua pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ol style="list-style-type: none"> 1) <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada

		<p>kasus berikut</p>  <pre> graph TD Login((Login)) -.-> <<include>> ValidasiUsername((validasi username)) </pre> <p>2) <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut :</p> <p>Kedua interpretasi di atas dapat dianut salah satu / keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>  <pre> graph TD UbahData((ubah data)) -.-> <<include>> ValidasiUser((validasi user)) </pre>
--	---	---

2. Class Diagram

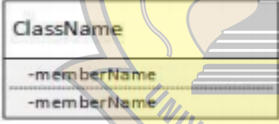

Menurut (Shalahuddin, 2015) diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian





kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut pola dan metode atau operasi :

- 1) Atribut merupakan variable-variable yang dimiliki oleh suatu kelas.
- 2) Operasi atau metode adalah fungsi yang dimiliki oleh suatu kelas

Tabel 3. 2 Simbol Class Diagram

Sumber : (Shalahuddin, 2015)

NO	SIMBOL	DESKRIPSI
1.	Kelas 	Kelas pada struktur sistem.
2.	Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi berarah/ <i>directed association</i> . 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

4.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi (umum-khusus).
5.	<p>Asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
6.	<p>. Kebergantungan / <i>dependensi</i></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	<p>Agragasi / <i>aggregation</i></p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

3. Activity Diagram


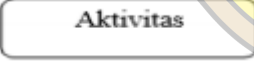


Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Shalahuddin, 2015). Menurut (Shalahuddin, 2015) Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :


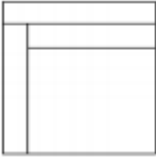
- 1) Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- 2) Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.

- 3) Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kamus ujinya.
- 4) Rancangan menu yang ditampilkan pada perangkat lunak.

Tabel 3. 3 Simbol activity diagram

Sumber : (Rosa dan Shalahuddin,2015:162)

NO	SIMBOL	DESKRIPSI
1.	status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.


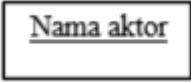

5.	<p>Status akhir</p> 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<p>Swimlane</p> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

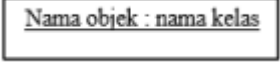


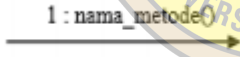
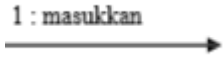
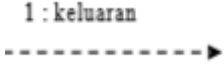
4. Sequence Diagram


Menurut Rosa dan Shalahuddin (2015: 165) diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

Tabel 3. 4 Simbol sequence diagram

Sumber : (Shalahuddin, 2015)

NO	SIMBOL	DESKRIPSI
1.	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam kata benda diawal frase nama aktor.
2.	<p>Garis hidup/<i>lifeline</i></p> 	Menyatakan kehidupan suara objek

3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
4.	<p>Waktu aktif</p> 	<p>Objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif adalah sebuah tahapan yang dilakukan di dalamnya.</p>
5.	<p>Pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
6.	<p>. Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p>
7.	<p>Pesan tipe send.</p> 	<p>operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi. Pesan tipe send suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
8.	<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek</p>

		yang menerima kembalian.
9.	Pesan <i>tipe destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i> .

3.5 Komponen Pada Website

3.5.1 PHP (*Hypertext Pre-processor*)

Menurut (Enterprise, 2017), mengemukakan “PHP merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *website*. Menurut (Andre, 2019), PHP adalah bahasa server-side *scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis.

3.5.2 HTML (*Hyper Text Markup Language*)

Menurut (Solichin, 2016) HTML merupakan bahasa pemrograman web yang memberitahukan pramban web (web browser) bagaimana menyusun dan menyajikan konten di halaman web.

Berdasarkan teori ini, maka *hyper text markup language* (HTML) merupakan bahasa pemrograman yang dikenal oleh *browser* untuk menampilkan informasi lebih menarik di halaman web melalui web *browser*.

3.5.3 CSS (*Cascading Style Sheet*)

Menurut (Koesheryatin, 2014) adalah suatu Bahasa pemrograman *web* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam. CSS atau *cascading style sheet* bahasa pemrograman yang diusulkan oleh Hakon Wilum Lie pada tahun 1994 dan distandarisasi oleh W3C yang berfungsi untuk mempercantik tampilan web (Solichin, 2016).

Maka dari itu, *Cascading Style Sheet* (CSS) merupakan bahasa pemrograman web yang digunakan untuk mengatur konten dalam sebuah halaman *web* yang ditulis dalam bahasa markup agar halaman web tersebut lebih menarik dan terstruktur.

3.5.4 Java Script

Javacript adalah sekumpulan perintah khusus yang digunakan untuk membuat sebuah halaman *web* yang lebih *responsif* dan *interaktif*. *Javascript* merupakan *script* yang dicantumkan pada sebuah halaman web dan dijalankan pada penjelajah web (*web browser*). *Javascript* terutama terkenal karena penggunaannya dihalaman *web* yang memberikan kemampuan tambahan pada HTML dengan mengizinkan pengekseskuan perintah di sisi *user* (penjelajah web), bukan diisi *server web* (Sianipar, 2015)

3.6 Pengertian MySQL.

Menurut (Jubilee, 2014), My Structured Query Language (MySQL) adalah sebuah perangkat lunak sistem manajemen basis data yang multithread, multi-user dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU GPL (*General Public License*)

3.7 Pengertian XAMPP

Menurut (Riyanto, 2015) XAMPP adalah sebuah software yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL yang dijalankan di komputer secara lokal. XAMPP berperan sebagai web server pada komputer. XAMPP juga disebut sebuah *Cpanel server* virtual, yang dapat membantu melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus online atau terakses dengan internet.

Menurut (Purbadian, 2016), berpendapat bahwa “XAMPP merupakan suatu *software* yang bersifat *open source* yang merupakan pengembangan dari LAMP (*Linux, Apache, MySQL, PHP* dan Perl)”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa Xampp merupakan tool pembantu pengembangan paket perangkat lunak berbasis open source yang menggabungkan Apache web server, MySQL, PHP dan beberapa modul lainnya di dalam satu paket aplikasi.

3.8 Pengertian Downtime

Downtime didefinisikan sebagai waktu menganggur atau lama waktu dimana mesin tidak dapat lagi dijalankan untuk beroperasi sesuai dengan yang diharapkan. Atau dengan kata lain *downtime* didefinisikan sebagai waktu yang diperlukan selama peralatan atau mesin tidak dapat digunakan atau mesin mengalami kerusakan (gangguan), sehingga mesin atau peralatan tidak dapat menjalankan fungsinya sesuai dengan yang diharapkan dengan baik (Fadrila, 2006). Beberapa unsur di dalam konsep *downtime* :

1) *Maintenance delay*

Maintenance delay merupakan waktu yang diperlukan untuk menunggu ketersediaan sumber daya *maintenance* yang akan melakukan proses perbaikan. Sumber daya *maintenance* dapat berupa teknisi, peralatan bantu, alat pengelasan, dan komponen pengganti.

2) *Supply delay*

Supply delay merupakan waktu yang dibutuhkan oleh personel *maintenance* untuk memperoleh komponen yang diperlukan dalam melakukan proses perbaikan. *Supply delay* terdiri dari *lead time* administrasi, *lead time* produksi, dan waktu transportasi komponen ke lokasi perbaikan.

3) *Acces time*

Acces time merupakan waktu yang dibutuhkan untuk memperoleh akses ke komponen yang mengalami kerusakan.

4) Diagnosis time

Diagnosis time merupakan waktu yang dibutuhkan untuk mengidentifikasi penyebab kerusakan yang terjadi serta mempersiapkan langkah - langkah yang diperlukan untuk memperbaiki kerusakan.



USM